Transformar Floating ActionButton en ToolBar





Juan Carlos Parralejo Martínez

1. Styles.xml.

El primer paso que tenemos que hacer, al trabajar con AppBar es añadir las siguientes lineas al archivo styles.xml, indicando que no usaremos la actionBar ni el título.

```
AndroidManifest.xml
                                             a colors.xml
                                                                           MainActivity.java
activity_main.xml
                                                            👼 styles.xml 🗡
                                                                                               app
       <resources>
            <!-- Base application theme. -->
            <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
                <!-- Customize your theme here. -->
5
6
                <item name="colorPrimary">@color/colorPrimary</item>
7
  <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
                <item name="colorAccent">@color/colorAccent</item>
                <item name="windowActionBar">false</item
                <item name="windowNoTitle">true</item>
10
            </style>
            <style name="AppCompatTheme" parent="@style/Theme.AppCompat.Light">
            </style>
            <style name="AppTheme.PopupOverlay" parent="ThemeOverlay.AppCompat.Dark.ActionBar">
14
15
            </style>
17
      </resources>
18
```

Si no cambiamos esto, la actionBar entra en conflicto con la Appbar.

2. Archivo build.gradle

En el archivo build.gradle de la aplicación(no del proyecto) alñadimos las siguientes lineas para añadir las dependencias para usar FABToolbarLayout y material design. Se hace de esta forma en la version 3.5 de Android Studio, ya qu no lo sugiere en Project Structure – Dependencies.

```
🔾 🛬 💠 — 👼 activity_main.xml × 🙀 AndroidManifest.xml × 🙇 colors.xml × 👼 styles.xml × 💿 MainActivity.java × 🔊 app ×
   ▼ 📭 app
                                                                                                                                  Open Project Structure Hide notification
                                           Configure project in Project Structure dialog.

    ■ manifests

                                                    apply plugin: 'com.android.application
           AndroidManifest.xml
                                                   android {
Resource Manager
                                                        compileSdkVersion 29
     ▶ 🗽 java (generated)
                                                        buildToolsVersion "29.0.0"
      ▼ 📭 res
                                                        defaultConfig {
        drawable
                                                            applicationId "com.example.ejerciciopropuestofab"
        ▼ 🛅 layout
                                                            minSdkVersion 28
              activity_main.xml
                                                            targetSdkVersion 29
ı.
                                                            versionCode 1
        ▼ 🖿 mipmap
                                                            versionName "1.0"
           ▼ 🖿 ic_launcher (6)
                                                            testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner
                ic_launcher.png (hdpi)
                ic_launcher.png (mdpi)
                                                        buildTypes {
                ic_launcher.png (xhdpi)
                                                            release
                ic launcher.png (xxhdpi)
                                                                minifyEnabled false
                ic_launcher.png (xxxhdpi)
                                                                proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'). 'proguard-rules.pro'
                ic_launcher.xml (anydpi-v26
           ▶ 🖿 ic_launcher_round (6)
        ▼ 🖿 values
             acolors.xml
                                                   dependencies {
              🗸 strings.xml
                                                        implementation fileTree(dir: 'libs', include: ['*.jar'])
                                          24
                                                        implementation 'androidx.appcompat:appcompat:1.1.0
             styles.xml
                                                        implementation \ 'and roidx. constraint layout: constraint layout: 1.1.3'
        res (generated)
                                          26

▼ M Gradle Scripts

        w build.gradle (Project: EjercicioPropues 28
        w build.gradle (Module: app)
                                                        testImplementation 'junit:junit:4.12'
        gradle-wrapper.properties (Gradle Vers 30
        proguard-rules.pro (ProGuard Rules fo 31
                                                        androidTestImplementation 'androidx.test.ext:junit:1.1.1'
        gradle.properties (Project Properties) 32
                                                        androidTestImplementation 'androidx.test.espresso;espresso-core:3.2.0
        settings.gradle (Project Settings)
        local.properties (SDK Location)
```

3. Archivo activity_main.xml

Es necesario que el contenedor principal sea un CoordinatorLayout, dentro de este, en primer lugar crearemos la ToolBar como hicimos en el tema 3, como un elemento independiente.

```
<com.google.android.material.appbar.AppBarLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:theme="@style/ThemeOverlay.AppCompat.ActionBar">
    <androidx.appcompat.widget.Toolbar
        android:id="@+id/toolbar"
        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize"
        android:background="?attr/colorPrimary"
        app:popupTheme="@style/Theme.AppCompat.Light"/>
```

El segundo componente es un FABToolbarLayout, el cual contiene el FloatingActionButton y el contenido de la ToolBar.

```
<com.github.fafaldo.fabtoolbar.widget.FABToolbarLayout
    android:id="@+id/fabtoolbar"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:containerId="@id/fabtoolbar_container"
    app:fabId="@id/fab"
    app:fabToolbarId="@id/fabtoolbar_toolbar"
    app:fadeInFraction="0.2"
    app:hideDuration="600"
    app:horizontalMargin="8dp"
    app:showDuration="600"
    app:verticalMargin="16dp">

<!-- <include layout="@layout/activity_main"/> -->
```

Con <u>app:containerId</u> hacemos referencia a la ToolBar, con <u>app:fabId</u> hacemos referencia al FAB y con <u>app:fab</u>ToolbarId hacemos referencia al contenido de la ToolBar.

Con la anotación app:fadeInFraction indicamos el procentaje de desplazamiento en la animación y con app:showDuration, la duración de esta.

Dentro del FABToolbarLayout meteremos el FAB y el contenido de la ToolBar.

```
<RelativeLayout
    android:id="@+id/fabtoolbar_container"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
    <com.google.android.material.floatingactionbutton.FloatingActionButton
        android:id="@+id/fab"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/add"
        android:backgroundTint="@color/azul"
        app:borderWidth="Odp"
        app:fabSize="normal"/>
    </RelativeLayout.</pre>
```

```
</RelativeLayout
</pre>

android:id="@+id/fabtoolbar_toolbar"
    android:layout_width="match_parent"
    android:layout_height="70dp"
    android:layout_alignParentBottom="true"
    android:orientation="horizontal">

    android:id="@+id/uno"
    android:layout_width="0dp"
    android:layout_height="match_parent"
    android:layout_weight="1"
    android:scaleType="centerInside"
    android:src="@drawable/account"
    />
```

4. Archivo Main Activity. java

En este creamos todos los Objetos del xml mendiante el método findViewById() y creamos los escuchadores de los botones implementando en la clase View.OnClickListener.

En el método onClick, mostraremos lo que hemos creado en en activity_main.xml con el método de FABToolbarLayout, show().

Tambien destacat el método setSupportActionBar que establece la barra de herramientas como la barra de app de la actividad.